14、系统睡眠唤醒--定时器唤醒

1．实验目的

1)　为什么要睡眠? Zigbee 的特点就是远距离低功耗的无线传输设备，节点模块闲时可以进入睡眠模式，在需要传输数据时候进行唤醒，能进一步节省电量。

2)　掌握几种系统电源模式的基本设置及切换。系统电源有以下几种管理模式：全功能模式，高频晶振（16M 或者 32M ）和低频晶振（32.768K RCOSC/XOSC ）全部工作, 数字处理模块正常工作。

PM1：高频晶振（16M 或者 32M ）关闭，低频晶振（32.768K RCOSC/XOSC ）工作，数字核心模块正常工作。

PM2 :低频晶振（32.768K RCOSC/XOSC ）工作, 数字核心模块关闭，系统通过 RESET,外部中断或者睡眠计数器溢出唤醒。

PM3：晶振全部关闭，数字处理核心模块关闭，系统只能通过 RESET 或外部中断唤醒。此模式下功耗最低。

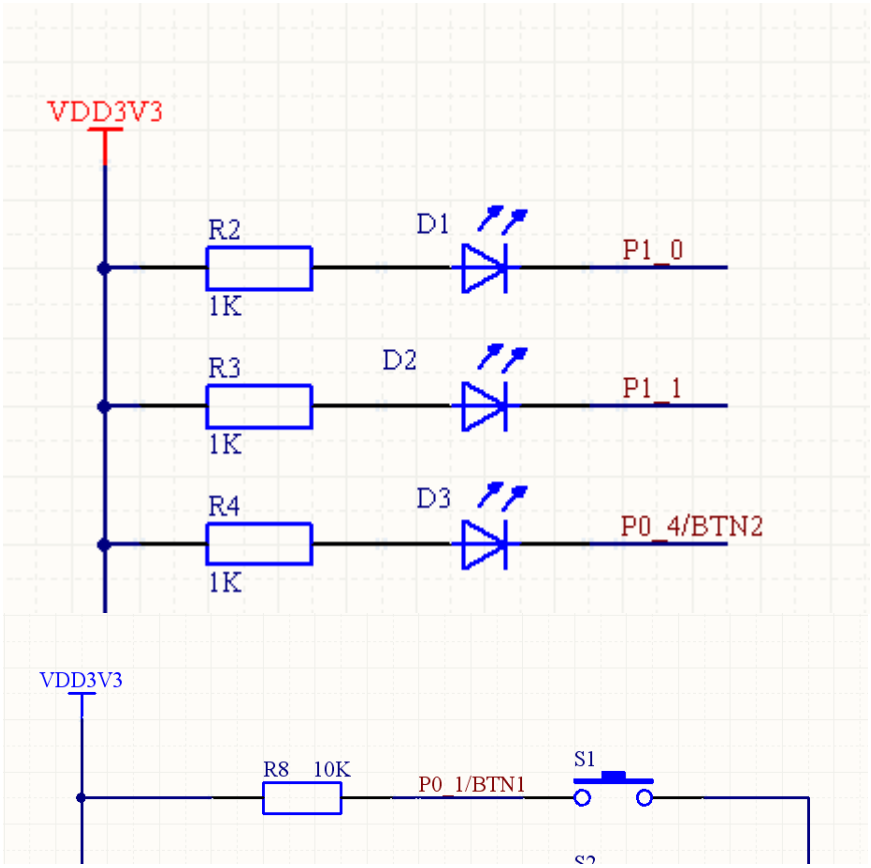
3)　将睡眠模式下的 CC2530 通过定时器唤醒，观察 LED 闪烁现象

2．实验设备

硬件：PC 机一台 ZB 网关（底板、核心板、仿真器、USB 线） 一套

软件：2000/XP/win7 系统，IAR 8.10 集成开发环境

3．实验相关电路图



由于发光二级管单向导电特性，即只有在正向电压（二极管的正极接正，负极接负）下才能导通发光。P1.0 引脚接发光二极管(D1)的负极,所以 P1.0 引脚输出低电平 D1 亮，P1.0 引脚输出亮电平 D1 熄灭。

4．实验分析及相关寄存器

相关寄存器 PCON，SLEEPCMD, ST0，ST1，ST2，如下表所示：（CC2530 中文数据手册完全版.pdf）

寄存器	作用	描述
PCON (0x87)	供电模式控制	Bit[0] 供电模式控制。写 1 到该位强制设备进入 SLEEP.MODE（注意 MODE=0x00 且 IDLE = 1 将停止 CPU 内核活动）设置的供电模式，这位读出来一直是 0。当活动时，所有的使能中断将清除这个位，设备将重新进入主动模式。
SLEEPCMD(0xBE)	睡眠模式控制	Bit[1:0] 供电模式设置 00 ： 主动/空闲模式 01 ： 供电模式 1 10 ： 供电模式 2 11 ： 供电模式 3
ST0		睡眠计数器数据 Bit[7:0]
ST1		睡眠计数器数据 Bit[15:8]
ST2		睡眠计数器数据 Bit[23:16]

睡眠定时器用于设置系统进入和退出低功耗睡眠模式之间的周期。还用于当系统进入低功耗模式后，维持MAC 定时器（T2）的定时。

其特性如下： 长达24 位定时计数器，运行在32.768KHZ的工作频率。24 位的比较器具有中断和DMA 触发功能在PM2 低功耗模式下运行.按照表格寄存器的内容，相关寄存器配置如下：

```
SLEEPCMD |= mode; //设置系统睡眠模式mode 取值为0、1、2、3
PCON = 0x01; //进入睡眠模式,通过中断唤醒
PCON = 0x00; //通过中断唤醒系统
```

5．源码分析

```
#include <ioCC2530.h>

#define uint unsigned int
#define uchar unsigned char
#define DELAY 15000
```

```
#define LED1 P1_0
#define LED2 P1_1      //LED 灯控制 IO 口定义

void Delay(void);
void Init_IO_AND_LED(void);
void SysPowerMode(uchar sel);

/*****************************************************************
   延时函数
*****************************************************************/
void Delay(void)
{
        uint i;
        for(i = 0;i<DELAY;i++);
        for(i = 0;i<DELAY;i++);
        for(i = 0;i<DELAY;i++);
        for(i = 0;i<DELAY;i++);
        for(i = 0;i<DELAY;i++);
        for(i = 0;i<DELAY;i++);
        for(i = 0;i<DELAY;i++);
        for(i = 0;i<DELAY;i++);
        for(i = 0;i<DELAY;i++);
        for(i = 0;i<DELAY;i++);
}

/*****************************************************************
系统工作模式选择函数
* para1  0    1       2       3
* mode       PM0   PM1   PM2   PM3

*****************************************************************/
```

```
void SysPowerMode(uchar mode)
{
  uchar i,j;
  i = mode;
  if(mode<4)
  {
    SLEEPCMD &= 0xFC;
    SLEEPCMD |= i;      //设置系统睡眠模式
    for(j=0;j<4;j++);
    {
      PCON = 0x01;        //进入睡眠模式
    }
  }
  else
  {
    PCON = 0x00;          //系统唤醒
  }
}

/*****************************************************************
     LED 控制 IO 口初始化函数
*****************************************************************/
void Init_IO_AND_LED(void)
{
    P1DIR = 0X03;
    LED1 = 1;
    LED2 = 1;
    //P0SEL &= ~0X32;
    //P0DIR &= ~0X32;
    P0INP  &= ~0X32;   //设置 P0 口输入电路模式为上拉/下拉
    P2INP &= ~0X20;    //选择上拉
```

```
    P0IEN |= 0X32;     //P01 设置为中断方式
    PICTL |= 0X01;     //下降沿触发
    EA = 1;
    IEN1 |= 0X20;      // 开 P0 口总中断
    P0IFG |= 0x00;     //清中断标志
};
/*****************************************************************
    主函数
*****************************************************************/
void main()
{
  uchar count = 0;
  Init_IO_AND_LED();
  LED1 = 0 ;       //开 LED1，系统工作指示
  Delay();         //延时
  while(1)
  {
    LED2 = !LED2;
    LED1 = 0;
    count++;
    if(count >= 6)
    {
      count = 0;
      LED1 = 1;
      SysPowerMode(3);
      //3 次闪烁后进入睡眠状态 PM3
    }
    //Delay();
    Delay();
    //延时函数无形参，只能通过改变系统时钟频率或 DEALY 的宏定义
    //来改变小灯的闪烁频率
```

```
  };
}
/****************************************
    中断处理函数-系统唤醒
****************************************/
#pragma vector = P0INT_VECTOR
 __interrupt void P0_ISR(void)
 {
  if(P0IFG>0)
  {
     P0IFG = 0;
  }

  P0IF = 0;
  SysPowerMode(4);
 }
```

6.实验现象

在工作情况下，LED1 灯长亮，LED2 闪烁三次后，系统进入睡眠。当按下 S1，系统唤醒进入
工作状态，LED1 灯长亮，LED2 闪烁三次后，再次进入睡眠。